

## 電磁石架台のリモート制御(2)

### CONTROL OF MAGNET SUPPORT FRAME (2)

牛本信二<sup>#A)</sup>, 榎本嘉範<sup>B)</sup>, 佐々木信哉<sup>B)</sup>

Shinji Ushimoto<sup>#A)</sup>, Yoshinori Enomoto<sup>B)</sup>, Shinya Sasaki<sup>B)</sup>

<sup>A)</sup> Mitsubishi Electric System & Service Co.,Ltd.

<sup>B)</sup> KEK

#### Abstract

A motor control system was integrated into the magnet support frame for pulsed magnets installed in 2017. All-in-one motor controller in 3U height rack mount case, which contains a power supply, stepping motor driving circuits, an Arduino with digital signal processing circuits and a raspberry pi for control and communication, was developed. One of the feature of the controller is that the EPICS IOC runs on the raspbian OS. Information from position sensors are also integrated in the system and used for feed-back control of the stepping motors. The system makes it possible for us to control the position of the magnet support frame with 10  $\mu\text{m}$  precision.

#### 1. はじめに

KEK 電子陽電子入射器(以下、入射器)では下流の蓄積リング(PF-RING、PF-AR、SuperKEKB HER/LER)へ 20ms 毎にビームのエネルギーを切り替えて入射[1]をおこなっている。

近年の高度化では、より高品質なビームを供給するために最適な光学系による輸送と、加速器コンポーネントの精密アライメントを実現するための改造を進めている。

光学系の最適化には、エネルギーが異なるビームラインの四重極マグネットとステアリングマグネットをパルスマグネットに交換した。これに合わせて、精密なアライメントが可能な電磁石用独立架台の開発[2]をおこない、2017年にパルスマグネットを搭載してビームラインに設置した。Figure 1 にビームラインへのパルスマグネットおよび架台の設置状況を記す。

この架台は、当初からモーター駆動による遠隔操作を想定して開発をおこなっており、2018年度の長期運転停止期間中に、全架台へのモーターの組み込みを完了した。また、制御用コントローラーを開発し、実際のビーム運転時において架台の遠隔制御をおこなった。

本稿では電磁石架台のモーター制御システムの概要と開発したコントローラーの詳細、運転時に使用する架台制御プログラムについて紹介する。

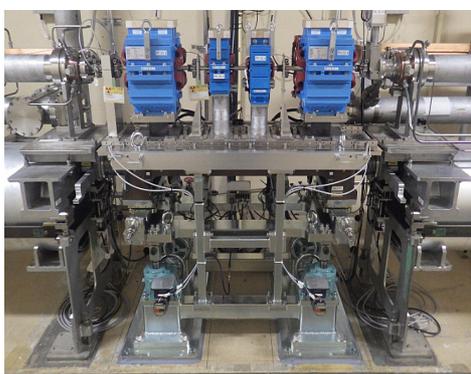


Figure 1: Pulsed magnet with support frame.

#### 2. モーター制御システム概要

モーター制御システムの概略図を Figure 2 に記す。

開発した架台には駆動部 6 軸(垂直方向:4 軸、水平方向:2 軸)にスクリージャッキが組み込まれており、これを操作することで、ビーム進行方向の並進を除く 5 方向への調整を実現する。このスクリージャッキにモーターを連結することで、架台の遠隔制御が可能となる。モーターは微調整が可能なステップモーターを使用する。

架台の調整量は可動部から独立した位置に取り付けたリニアゲージ(ミツヨト製 LGS-1012P)で測定し、その値を用いてモーターの制御をおこなっている。このゲージの分解能 (10  $\mu\text{m}$ ) がそのまま架台の調整精度となる。

架台にはビームライン保護のための、各軸の調整限界位置にリミットスイッチが取り付けられている。架台がリミットスイッチに当たると、モーターコントローラー回路内のリレーで強制的にモーターの出力を遮断する仕組みとなっている。

機器の制御をおこなうモーターコントローラーは地上部の筐体に配置し、そこからトンネル内へは多芯ケーブル 1 本で中継 I/F BOX へ配線することで、トンネルギャラリ間の配線数を減らしている。また、I/F BOX から機器への配線もコネクタ付きのケーブルを使用することで、機器交換時のケーブル接続コストを低減している。

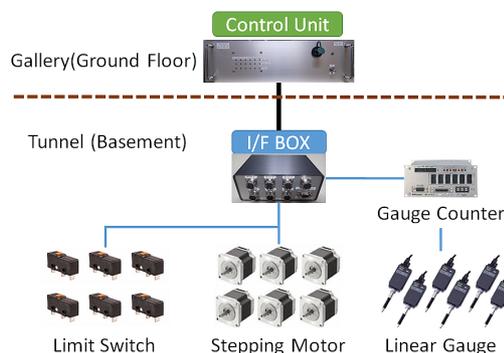


Figure 2: Overview of motor control system.

<sup>#</sup> ushimoto@post.kek.jp

### 3. モーターコントローラー

架台のモーター制御に使用するコントローラーは、EPICS[3]をはじめ、システムに使用するリニアゲージやリミットスイッチとモーター制御部をコンパクトな筐体に統合することを目標に、独自に開発をおこなった。また、このコントローラーは電磁石架台だけでなく、最大 6 軸のモーションコントローラーとしての使用も想定した作りとなっている。Figure 3 に開発したコントローラーを記す。

以下にコントローラーに搭載した機器と、架台に設置したリニアゲージ用カウンター(EV-16D)の詳細について紹介する。

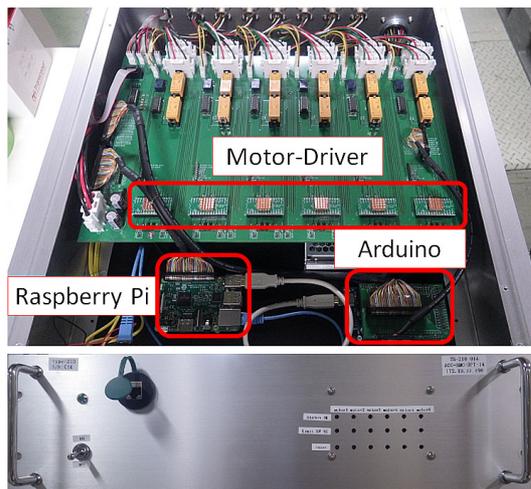


Figure 3: Motor Control Unit.

#### 3.1 Raspberry Pi

コントローラーの主な制御は Raspberry Pi 3 Model B[4]を使用した。このマイコンは 1.2GHz、4quad core の ARM プロセッサを搭載し、Linux ベースの Raspbian OS で動作する。Raspberry Pi は安価でありながら、様々なデジタル通信(シリアル、I2C、SPI)を備えている。また、10/100 Mbps イーサネット、USB2.0 4ポート、WiFi (2.4GHz)、Bluetooth 4.1 などの豊富な I/F を有し、General Purpose Input/Output (GPIO) と呼ばれるピンヘッダが搭載されている。Figure 4 に開発したコントローラーで使用した GPIO ピンヘッダを記す。

本コントローラーでは、arduino および EV-16D との間で、シリアル通信をおこなっている。それぞれ USB、ピンヘッダの 8、10 番ピンにアサインされている GPIO 14 (TxD)、GPIO 15 (RxD) を介して接続されている。

一方、モータードライバーとの通信には SPI(Serial Peripheral Interface) を使用する。標準で搭載された SPI 用 GPIO のチップセクタ(CS)は 2 系統のみであるため、そのままでは 6 台のモータードライバーを制御できない。そこで標準の MOSI/MISO/SCLK ピンと、汎用 GPIO(21~26) を標準 CS ピンの代わりに組み合わせることで、6 台のドライバーを制御している。

また、Raspberry Pi の電源は通常 micro USB もしくは AC アダプタで給電をおこなうが、ピンヘッダの 2 番ピン(5V PWR) へ外部からこの給電することで電源として使用している。

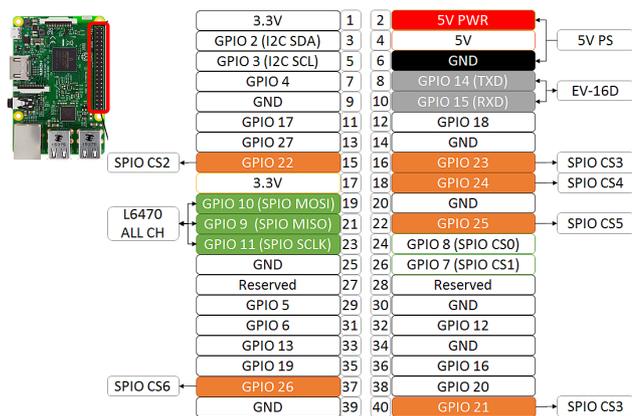


Figure 4: Layout of GPIO for Motor Controller.

#### 3.2 Arduino

コントローラー内で使用するデジタル I/O は、Raspberry Pi の GPIO だけでは不足するため、別途 Arduino Mega 2560[5]を使用している。この Arduino は I/O ピンが 54 本あり、チャンネル毎に以下の入出力として使用している。

- モータードライバー IC の信号入力
- リミットスイッチ接点入力
- モータードライバー出力用リレーモニタ信号
- 出力用リレー回路のリセットスイッチへの出力

実際にはこれに予備を含めた、計 30 点の信号をハードウェア側で処理し、コントローラー表側のステータス用 LED を点灯させている。一方、ソフトウェアでは、独自の Arduino 言語を用いて作成したシリアル通信用のプログラミングが、USB で接続した Raspberry Pi との間でデータの送受信をおこなっている。

#### 3.3 ステッピングモータードライバー

ステッピングモータードライバーには秋月電子のモータードライバーモジュール AE-L6470DIP28[6]を使用した。このモジュールは 2 相バイポーラスステッピングモーター制御 IC (L6470)を使用している。このドライバーはモーター制御のためのパラメータ設定用コマンドが用意されており、SPI 通信でこのコマンドを送信することで制御をおこなう。

本システムでは、実際にマグネットを搭載した試験架台を使用し、モーターの動作や、IC の発熱などを観測したうえで、稼働時および制止時の励磁電流を決定している。また、実際の架台調整では、回転速度を変化させて制御をおこなっている。

#### 3.4 EV-16D

架台に設置したリニアゲージ用の 6ch カウンターであり、各架台に 1 台ずつ設置している。地上のコントローラーと接続されており、給電もコントローラーからおこなう。

このカウンターはゲージの値を直接表示する表示部は搭載されておらず、RS-232 用インターフェースを介したシリアル通信で値の取得をおこなう。

2018 年に設置以降、ビームライン直下でリニアゲージと共に使用しているが、現時点で故障は発生していない。

## 4. EPICS による制御

入射器では EPICS を中心とした制御システムが構築されており、様々な機器を Channel Access プロトコルを介して制御する。開発したコントローラーでは Raspberry Pi を EPICS IOC (Input/Output Controller) として使用する。Table 1 にコントローラーの EPICS 環境を記す。

Table 1: EPICS Environment for Raspberry Pi

EPICS Base & Modules	Version
Base	R3.15.5
Seqencer	2-2-5
asynDriver	4-32
StreamDevice	2-7-7
Devgpio	R1-0-5
drvAsynSPI	-

### 4.1 GPIO 制御

開発したコントローラーでは Raspberry Pi の GPIO を使用する。EPICS レコードから直接 GPIO の制御をおこなうため、devgpio[7]を作成利用した。このドライバーサポートは仮想ディレクトリ配下のファイル“/sys/class/gpio /GPIO(xx)/Value” へ対して読み書きをおこなうことで GPIO の制御をおこなう。レコード型は bi/bo をサポートしており、EPICS データベース内で DTYP(“devGpio”) を選択、IN/OUT field にピン番号および値(H,L)を指定して使用する。

なお、このデバイスドライバは、Raspberry Pi だけでなく、BeagleBone Black でも利用できる。

### 4.2 モータードライバー制御

モータードライバーとの SPI 通信では、標準 SPI 出力ピンと CS 用に汎用 GPIO を併用している。SPI ピンは標準 SPI 入出力を制御するために drvAsynSPI[8]を利用している。

実際の制御では的確なタイミングで CS 用 GPIO の処理をおこなう必要があるため、以下の順序となるように、EPICS データベース内で各処理用レコードを FLNK で接続している。

- (1) CH 指定レコード(CH[X]:SET) 確認
- (2) 指定 CH の CS 用 GPIO 切り替え(H→L)
- (3) SPI コマンド送受信 (MISO/MOSI)
- (4) 指定 CH の CS 用 GPIO 切り替え(L→H)

この方法では、選択した複数のモータードライバーへ同時にコマンドを送信することも可能である。

### 4.3 シリアル通信

Raspberry Pi と Arduino およびリニアゲージカウンター(EV-16D)は AsynDriver と StreamDevice を用いて、シリアル通信によるデータの送受信をおこなう。前者は USB 経由(/dev/ttyACM0)、後者は GPIO のシリアルポート(/dev/ttyAMA0) を介して機器と接続する。GPIO によるシリアル通信を使用する場合、本システムで使用

した Raspberry Pi 3 model B では、/dev/ttyAMA0 が Bluetooth 用にアサインされており、GPIO によるシリアル通信は使用できない。そのため、本システムでは使用しない Bluetooth 機能を OFF することで、GPIO からのシリアル通信を使用している。

### 4.4 シーケンス制御

EPICS IOC 上では、モーターの制御および各種モニタ(リニアゲージ読み出し、モータードライバステータス、リミットスイッチ状態)信号に対して個々のレコードを作成し、Seqencer を使用して、制御シーケンスを構築している。Seqencer は IOC 上で動作するシーケンス制御用プログラムである。

実際の制御シーケンスは以下の通りである。

- (1) リニアゲージ値の健全性確認。異常時はカウンターへエラークリアを送信
- (2) 軸毎の設定値と現在値の比較。モーター回転中の場合、設定値との差分が 10  $\mu\text{m}$  以下の場合、ドライバーにモーター停止信号を送信
- (3) 軸選択用レコードのチェック。選択されていない場合は(1)へ戻る
- (4) (2)の差分を基にモーター回転速度を算出
- (5) (3)で選択した軸のドライバーに信号送信
- (6) 指定軸のモーター回転開始、(1)に戻る

実際の動作では、設定値に近づくにつれてモーターの回転速度が緩やかになり、設定値の $\pm 10 \mu\text{m}$  以内に入ったところでモーターが停止する。このように、本システムではリニアゲージの分解能に合わせて架台制御の精度が決まる。

実際の架台調整で使用するレコードは、各軸調整用のレコードを組み合わせることで、架台全体の昇降や水平方向の平行移動、XYZ 軸毎の回転(ピッチ、ヨー、ロール)をおこなう。なお、各軸は Fig. 5 に示した座標系を基に定義する。

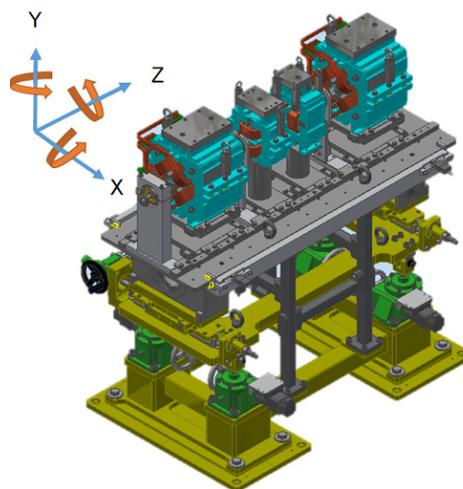


Figure 5: Coordinate System of Support Frame.

## 5. 架台調整プログラム

構築したモーター制御システムでは、整備した設定用 PV に値を書き込む(caput)ことで、容易に電磁石架台の

調整が可能となった。さらに効率よく架台制御をおこなうための GUI プログラムを LabVIEW を用いて作成した。Fig. 6 に作成した GUI を示す。LabVIEW からレコードへのアクセスは Ca Lab[9]を使用している。

パネルの左側には架台の平面図を配置し、その上に実際の調整箇所と対応したボタンを配置してある。これにより、直感的に架台の調整がおこなえる。

一方、右側には複数のプロットを配置している。上部はリニアゲージのトレンドモニター（水平方向、垂直方向）、下部は架台の各軸における回転状況のモニター（ロール、ヨー、ピッチ）を表示している。

これらのモニタにより、架台の状態をより詳細に把握しながら調整をおこなうことが可能である。

2018 年の運転では、実際にこのプログラムを使用して架台調整をおこなっている。ビーム軌道を一定に保った状態で架台を動かし、架台上に搭載したビーム位置モニター(BPM)でビーム位置の観測をおこない、架台の調整量に対してビーム位置が同じ量変化することを確認している。



Figure 6: Magnet Support Frame Control Program.

## 6. まとめ

2017 年に設置した電磁石架台のモーター制御システムの整備をおこなった。システムの中心となるモーターコントローラーは 6 軸制御に使用する複数の機器を 1 つの筐体に統合した独自のものを開発した。このコントローラーは、機器の制御用に Raspberry Pi を搭載し、それを EPICS IOC とすることで、入射器の制御システムから直接操作できる

このモーターコントローラーは、架台に取り付けたリニアゲージの測定値を用いてステッピングモーターを制御する。本システムの導入により、遠隔から 10  $\mu\text{m}$  の精度で架台を調整することが可能となった。

## 参考文献

- [1] M. Sato, “KEK 電子陽電子入射器制御システム”, Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31- Aug.3, 2019.
- [2] Y. Enomoto *et al.*, “電磁石架台のモーター制御 1”, Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31- Aug.3, 2019.
- [3] Experimental Physics and Industrial Control System;

- [4] <https://epics.anl.gov/raspberry-pi-3-model-b/>;  
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [5] Arduino Mega 2560;  
<https://store.arduino.cc/usa/mega-2560-r3>
- [6] AE-L6470DIP28;  
<http://akizukidenshi.com/catalog/g/gM-06365/>
- [7] devgpio; <https://github.com/ffeldbauer/epics-devgpio>
- [8] drvAsynSPI; <https://github.com/kek-acc/drvAsynSPI>
- [9] CA Lab;  
[https://www.helmholtz-berlin.de/zentrum/locations/it/software/exsteuer/calab/index\\_en.html](https://www.helmholtz-berlin.de/zentrum/locations/it/software/exsteuer/calab/index_en.html)